KAHENI PETER

P101/1740G/21

COM 405E: ARTIFICIAL INTELLIGENCE PROGRAMMING

COURSE WORK PROJECT

DOCUMENTATION

QUESTION

E-commerce Price Optimization (Retail)

Question: Build a model-based agent for an e-commerce platform that adjusts product

prices based on market demand, competitor prices, and user behavior to maximize

revenue.

Requirements:

1. Perceive data on demand trends, competitor pricing, and sales data.
2. Model demand elasticity and competitor influence on sales.
3. Decide on optimal pricing adjustments for different products.
4. Update prices in real-time and refine the model with new sales data.

DOCUMENTATION

**Overview**

**Project Description**

-The **E-Commerce Price Optimization System** is a machine learning-based application designed for e-commerce platforms.

-It dynamically adjusts product prices by analyzing market demand, competitor pricing, and user behavior.

- The system aims to maximize revenue through real-time price optimization.

## ****Project Objectives****

1. **Data Perception** - Gather data on demand trends, competitor prices, and sales history.
2. **Elasticity Modeling** - Build models to evaluate demand elasticity and competitor influence on sales.
3. **Optimal Pricing Decisions** - Implement algorithms to recommend the best pricing strategies for different products.
4. **Real-Time Integration** - Update prices dynamically and refine the model using new sales data.

**Features**

1. **Dynamic Price Adjustments** - Prices are optimized in real-time based on demand and competitor analysis.
2. **Elasticity Modeling** - Models predict how price changes affect customer demand and revenue.
3. **Real-Time Data Integration** - System fetches live sales and competitor data to ensure accurate adjustments.
4. **Continuous Model Improvement** - New sales data automatically retrains the model, ensuring it evolves with changing market conditions.

**System Workflow**

**1. Data Collection**

* **Input:** Sales history, competitor pricing, customer behavior and demand market trend data.
* **Sources:** I have created the datasets manually using python.

**2. Preprocessing**

* Clean and prepare data for modeling using techniques like handling missing values, scaling, and encoding.

**3. Model Training**

* I have used Random Forest Regressor.
* It is part of the **Pipeline** and serves as the main regression model for predicting the target variable (product id).

**Preprocessing Techniques:**

* **Standard Scaler**:
  + Used to scale numerical features (Transaction Amount, competitor price, and competitor discount) to ensure they are on a similar scale.
* **One Hot Encoder**:
  + Encodes categorical features (competitor location, product category) into numerical values, making them usable by the model.

.

**4. Real-Time Updates**

* Fetch live data periodically via APIs and re-optimize prices using the trained models.

**5. Decision Output**

* Recommend optimal price adjustments or promotions for each product.

**Tech Stack**

1. **Backend:**
   * Flask for web framework.
   * SQLAlchemy for database interactions.
2. **Frontend:**
   * HTML, CSS, JavaScript for user interface.
   * Bootstrap for responsive design.
3. **Machine Learning:**
   * Scikit-learn and TensorFlow for model development.
4. **Database:**
   * PostgreSQL/MySQL for storing sales and competitor data.
5. **APIs:**
   * Integrated APIs for live data on competitor pricing and user behavior.
6. **Visualization:**
   * Matplotlib and Plotly for trend analysis and interactive dashboards.

**Key Algorithms**

1. **Demand Elasticity Modeling**
   * Regression models analyze the relationship between price and demand.
2. **Competitor Influence**
   * Feature engineering evaluates how competitor prices impact sales.
3. **Real-Time Forecasting**
   * Time-series models predict future trends in demand and pricing.
4. **Optimization**
   * Optimization algorithms recommend price adjustments that maximize revenue.

**Setup Instructions**

**1. Prerequisites**

* Python 3.8 or higher installed.
* PostgreSQL/MySQL installed.
* Required libraries:

pandas==2.0.3

numpy==1.25.2

scikit-learn==1.3.1

matplotlib==3.8.1

seaborn==0.13.0

xgboost==1.7.6

lightgbm==3.3.5

flask==3.1.0

joblib==1.3.2

statsmodels==0.14.0

scipy==1.11.3

mlxtend==0.21.0

pytest==7.4.2

SQLAlchemy==2.1.0

psycopg2-binary==2.9.7  # For PostgreSQL

mysql-connector-python==8.1.0  # For MySQL

**2. Installation**

1. Clone the repository
2. Set up the database: - Import the schema (schema.sql) into your database.
3. Start the Flask application

**3. Accessing the Application**

* Open your browser and go to <http://localhost:5000>.

**Usage Instructions**

1. **Input Data**
   * Enter values for transaction amount, competitor price, competitor discount, and product category.
2. **Select Competitor Location**
   * Use the dropdown to select the competitor's location.
3. **Get Predictions**
   * Click the **Get Prediction** button to view recommended price adjustments and strategies.
4. **Analyze Trends**
   * Navigate to /dashboard to view sales trends and performance over time.

**Future Enhancements**

1. **Advanced Predictive Models**
   * Incorporate deep learning models for more accurate demand predictions.
2. **Scalability**
   * Support for multi-regional price optimization with location-specific data.
3. **User Experience**
   * Enable personalized recommendations for customers.
4. **Reporting Tools**
   * Add functionality to export predictions and trends as CSV or Excel files.

**Contributors**

* **Name:** KAHENI PETER
* **Email:** kaheni.kinyanjui@s.karu.ac.ke
* **GitHub:** Not yet created

## ****References****

1. Flask Documentation: https://flask.palletsprojects.com/
2. TensorFlow Documentation: <https://www.tensorflow.org/>
3. PostgreSQL Documentation: <https://www.postgresql.org/>